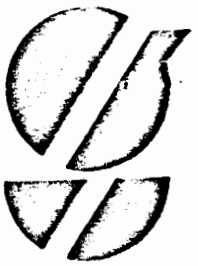

Gemini Microcomputers

FDC CARD

80-BUS Floppy Disk Controller Card

SOFTWARE MANUAL

CP/M 2.2 for Nascom Configurations

Gemini Microcomputers
Oakfield Corner,
Sycamore Road,
Amersham, Bucks HP6 5EQ

Issue 1 For Part No G513 27-8-81

CONTENTS

1. SCOPE

This manual is intended as a companion to the normal CP/M manuals, it does not attempt to teach the user about CP/M or how to use the standard CP/M software. It covers the implementation of CP/M for a Nascom using the Gemini Disk controller card. As such it concerns itself mainly with the BIOS and the features included therein. It also covers the two Gemini-supplied routines FORMAT and BACKUP. It is assumed that the CP/M manuals will be read in conjunction with this one. For those totally unfamiliar with CP/M a book such as

The CP/M Handbook by Rodney Zaks

will be found useful as the CP/M manuals are not as clear as they might be. As with most things with a high technical content, it may take several readings of the manuals before everything makes sense.

2. GENERAL

The supplied software consists of an EPROM containing a simple monitor (SIMON) and disk loader program, and a 5.25" disk containing your distribution copy of CP/M 2.2. The disk already holds a working 32k system on its system track configured for a standard Nascom. The supplied BIOS will support up to four drives.

The EPROM contains a simple monitor (see Appendix 1) and the necessary loader routine to start up CP/M. The purpose of the monitor is to provide some debugging facilities in the event of problems with the hardware, but hopefully it will never be used. The EPROM resides at address 0F000H to 0F3FFH and assumes the Nascom screen is at 0F800H to 0FBFFH and workspace from 0FC00H to 0FFFFH. It will also support a Gemini Intelligent Video Controller Card (IVC).

As a reasonable amount of space is available on the system track a comprehensive BIOS has been supplied. This has various features included in it which are highlighted in the following sections. (eg See Disk error handling - section 2.6).

2.1 System start up

On reset the monitor clears the screen, puts up a sign-on message, selects Drive A and attempts to read in the first sector of the first track of the disk in that drive. This sector should hold the bootstrap loader that will load in the entire CP/M system from the remainder of the system track. If it is unable to read the sector, or the disk does not contain a CP/M system, an error message is displayed and the monitor proper is entered and the prompt ">>" appears on the screen. Once a correct disk has been inserted in drive A and the door of the drive closed, the process can be repeated by typing "B" for Boot.

Once control has been passed to the loader the monitor is no longer used. If for any reason the loader cannot successfully load the system it Halts after first filling the screen with exclamation marks. Reset has to be pressed to get the processor out of the Halted state. In the unlikely event of this occuring try again, it may only be a soft error - the small size of the Boot does not allow for a sophisticated in-built

retry procedure. If it persists try another disk before investigating the hardware.

Once the CP/M system has been loaded the screen will be cleared and a new sign-on message will appear giving the current system size. This will be followed by the CP/M prompt "A>".

NOTE:
Once the monitor has successfully read the boot sector into memory the monitor is no longer required. In fact it is permissible for the loader to load the CP/M system over the monitor (which in this case must obviously be in RAM). As a result it is possible to run a 62k CP/M system on a standard Nascom, or a 64k system on a Nascom using the Gemini IVC card. This is why, in the event of a loading error, the loader Halts and does not return to the monitor as it may have partially overwritten it. For a possible way of getting the monitor into RAM see the INMC 80* newsletter Issue 4.

2.2 Disk format

The software supports two disk formats, one single density and one double density. The two may be used together, but note that the disk in drive A must always be double density.

2.2.1 Single Density

The single density format is the same as that supported by the Gemini G805 disk system (SD Systems format, but extended for double-sided disks). The disk is double sided with 35 tracks per side, each track being divided into 18 sectors of 128 bytes each. The software uses first side 0 of the disk, then side 1. ie CP/M sees a disk with 70 tracks and 18 sectors per track. The first three tracks are reserved as system tracks and the directory area has room for 64 directory entries. (Having logged in a single density disk use the command STAT DSK: to confirm this - see your CP/M manual).

2.2.2 Double Density

The double density format uses a physical sector size of 512 bytes. By using a larger sector size than 128 it has been possible to increase the storage capacity of a disk. Each track of the disk holds 10 sectors. To minimise head movements and to increase system performance the software accesses first side 0 of the disk on a particular track, and then side 1 before stepping on to the next track. Therefore the disk can be regarded as having 35 tracks with 20 512-byte sectors per track. However due to the blocking/deblocking (see on) CP/M sees a disk of 35 tracks with 80 128-byte sectors per track. One track has been reserved as a system track, and because of the capacity of the disk (340k) the directory size has been set at a maximum of 128 entries. (Once again use STAT DSK: to see this).

---

*INMC 80 - International Nascom Microcomputer Club
c/o Oakfield Corner,Sycamore Road, Amersham, Bucks.

2.3 Auto Density

As mentioned above the software supports two disk formats, and the determination of which the system is to use for a particular drive is entirely automatic. However there is one small restriction and this is that the system disk in drive A must be double density. The disks in drives B,C and D may be of either density. The density of a disk in a drive is determined the first time that drive is accessed. This density type remains in force for that drive until a warm start is performed. Most programs terminate with a warm start (STAT is an example of an exception) and typing ^C has the same effect.

2.4 Blocking/Deblocking

All CP/M software transfers data to and from the disk in 128-byte 'chunks'. This is due to the fact that this was the sector size on the machine that CP/M was originally written for, (and is also a widely used IBM standard). It is only now with new technology and increasing packing densities that larger sector sizes become more attractive. In order to achieve this and still be compatible with previous CP/M software, (and also to allow programs to maintain economical 128-byte buffers rather than larger ones), some software is interposed between CP/M and the disk drivers. This software maintains a physical sector buffer in memory (512 bytes in size in our case) through which all the CP/M data transfers are passed. Those who are interested in the inner workings of this software are referred to the CP/M 2.2 Alteration Guide, section 12 and Appendix G. As a result of this blocking/deblocking some strange effects may be noticed. For example if a double density disk with no files on it, or only one or two, is inserted and DIR typed, CP/M will access the disk and list the files (or type NO FILE). If the disk is then changed for another and DIR typed again the disk will not be accessed, but the same list of files as before will appear. This is because in response to the second DIR command when CP/M goes to read the directory track the blocking /deblocking software finds that the buffer already holds the first sector of track 1, (which holds four CP/M sectors in our case), and so it has no need to read the disk as the buffer already holds the wanted data. (If the first disk had had more directory entries then the software would have read more sectors from the disk, and in starting again the track/sector would not have matched the buffer and so the new disk would have been read). To guard against this it is always advisable to type ^C whenever a disk is changed.

2.5 Sector skewing

In the literature on CP/M you may come across a mention of "sector skew". The reason for this and how it is achieved is explained below.

If CP/M accessed the sectors of a track sequentially the response of the disk system would be extremely slow. The reason for this is that often after one sector of the disk has been read the controlling program spends a short time processing before it reads the next sector. In the simplest case BDOS will be calculating the next sector of the file and setting up the disk drivers for the next transfer. An example of a longer process is the warm boot, when the reloaded CP/M system proceeds to read the directory of the disk on the logged-in drive. After reading a sector it proceeds to extract the disk allocation information from the directory entries in that sector before reading the next sector, a

process which can take some milliseconds to complete. By the time the next transfer is requested the read head of the drive will be well past the header record of the next sector, if not several more sectors as well. If the next wanted sector was the one that physically followed the previous one, then a delay of almost one complete disk revolution would occur before the requested sector could be read/written. To get round this problem CP/M utilises a sector translation routine within the BIOS. This uses a look up table to translate a logical CP/M sector number to a physical sector number on the disk surface. The translation table for example may start 1,4,7,10... which would mean that the software has the time it takes two sectors to pass the drive head available to it before it will miss the next wanted sector and have to wait.

The time delay between disk accesses, (and hence skew required), is application dependent, and CPU clock rate dependent (the slower the clock rate, the longer the processor takes). So rather than selecting a compromise value for the Gemini format, or one appropriate to a 2MHz processor, no translation at all is done for double density. (Single density is however, in order to be compatible with the SD Systems format). To obtain a skew in double density the disk is formatted with the sectors in a jumbled order, rather than being in sequential order, so that when the sectors are read/written in numerical order, a delay of several physical sectors can occur before the next wanted sector is found. This can be done by the format program as sectors are identified on the disk not by their physical position in relation to the index hole, but by the sector number written into the sector headers by the format program. For example if the Format program is told to use a skew of two it writes the sector numbers on the disk in the following order:- 0,4,7,1,5,8,2,6,9,3. From this it can be seen that having accessed one sector, two other sectors must pass under the head before the sector with next sequential sector number is found.

By taking this approach a skew factor can be chosen to suit an individual system, and yet the disk can be successfully read on any other Gemini system. This is because the skewing is achieved by placing the sectors in different positions on the disk while still maintaining physical sector numbers (as read by the 1797 controller integrated circuit) that match the logical sector numbers.

[By contrast the table translation method with a skew of two would access the physical sectors in the order 0,3,6,9 etc, and one with a skew of three in the order 0,4,8, etc].

2.6 Disk error handling

The BIOS traps all disk errors and reports them to the user specifying the drive, track and sector they occured on together with the type of error that occurred. Three options are then offered:- a) Retry (note up to eight retries may have already been attempted). b) Return the error to CP/M (in which case a BDOS error message will appear). c) Abort and restart by typing ^C.

Possible error messages are:-

Not Ready — Occurs on already logged in drives if the door is open (or there is no disk in the drive) when the software tries to read a disk in that drive. (Note this message only appears

after the motors have timed out). If desired a disk can be inserted and/or the door closed and "Y" typed in order to continue.

Disk Write Protected. Occurs when an attempt is made to write to a disk with a write-protect tab fitted. Either remove the tab and type "Y" to continue, or ^C to abort.

Write Fault This message is caused by a signal from the disk drive itself. Where Pertec FD250s are used this message should never appear.

Record Not Found. Occurs when the controller has been unable to locate an error free header record containing the correct track and sector numbers, or it has been unable to locate the following data block. If it is possible to read the disk after several attempts of typing "Y" in response to the retry? request it is advisable to copy the disk to another one and to reformat the one giving errors. If the error persists the information in that sector will have been lost.

CRC Error A CRC error has occured in a data block. For comments see above. If it is a source file that is being read it may be possible to ignore the error (type "N" in response to retry? and <return> in response to the BDOS error message) and, if it is a minor one, find it and correct it within your editor.

Lost Data This should not occur. If it does it implies that the CPU clock rate is too slow. The minimum CPU clock rate that the software can run with is 2MHz without any Wait States.

No disk/Wrong format This occurs when the software is unable to determine the density of the disk in the selected drive. The density is determined by trying to perform a "Read Address" command on that drive in first double density and then single density. This command will fail to complete if there is no disk present, or the disk is incorrectly formatted. Insert a correct disk (or close the drive door) and type "Y" to continue.

3. MOVCPMs

The disk holds two copies of the MOVCPM program. One is MOVCPMN.COM, the other MOVCPMV.COM. The former holds the BIOS appropriate to a standard Nascom with its memory mapped display, the latter has a BIOS which supports the Gemini IVC in place of the Nascom screen. It will also support the use of a Gemini keyboard on the IVC card. Use the one appropriate to your hardware configuration.

The system track of the distribution disk holds a 32k system for a standard Nascom without IVC card. Both MOVCPMs have the system

parameters set up for a 4MHz CPU clock rate, and will handle up to four drives connected to the disk interface card.

4. INPUT/OUTPUT

4.1 Keyboard Routine

This provides the usual facilities (auto repeat, blinking cursor etc) and runs in a conventional typewriter mode (ie shift gives upper case). Nascom 1 owners can use the @ key as a control key, and shift has to be used in conjunction with @ in order to generate the @ character. (Nascom 2 owners don't forget that you can also use the @ key as a control key - it is conveniently placed for typing control/P and control/Enter).Unlike the keyboard routine in Nas-Sys, control/A and control/a both return the code 01. ie typing control/(letter) returns the correct control character irrespective of the state of the shift key.

Two flags have been built into the keyboard routine. These control the way the routine handles alpha characters and the bracket characters (). Two unique key combinations have been selected to "toggle" these flags. These key combinations would not occur in normal use and so the keyboard can still generate all the normal control codes. The key combinations and their effect are listed below:-

Control/Enter Toggles the "Upper Case Lock" flag. This allows the alphabetic keys of the keyboard to be locked into an upper case mode, in which upper case letters are always returned irrespective of the position of the shift key.

Control/Backspace Toggles the "Bracket Exchange" flag. Normally the shift/8 and shift/9 keys return the round bracket pair (). With the "Bracket Exchange" flag set square brackets [] are returned instead.

The later combination will only be of interest to Nascom 1 owners as the square brackets are already available on the Nascom 2 keyboard. Square brackets are occasionally required in CP/M (for example see the PIP command in your CP/M manual).

On the Nascom 2 keyboard the extreme right-hand key (CH) now returns the code 09 (control/I) and so can be used as a TAB key when using any of the CP/M editors.

4.2 CRT Routine (For the IVC card refer to the IVC software manual).

Unlike Nas-Sys the CRT routine treats the display as 16 lines of 48 characters rather than 15 lines of 48 characters. ie The top line of the display is not stationary, but is scrolled along with the rest of the screen.

The control codes recognised by the CRT routine are listed below:

08 BACK Backspace. The cursor is moved one position to the left and the character now under the cursor is overwritten with a space.

0A LF Linefeed. The cursor is moved down one line on the display. If it is already on the bottom line the display is scrolled up by one line and the bottom line is cleared.

0D RETURN Carriage return. The cursor is returned to the start of the line in which it is currently positioned.

16 ^V Delete character from line. The character currently under the cursor is deleted, and the remaining characters on the line are shifted left one position. A space is entered in the last character position of the line.

17 ^W Insert character in line. A space is inserted at the current cursor position. The character under the cursor, and all characters to the right of it, are shifted one character position to the right. The character at the end of the line is lost.

19 ^Y Cursor Home. The cursor is returned to the start of the top line of the display.

1A ^Z Clear Screen. The entire display is cleared and the cursor is moved to the start of the top line.

1B ^[Escape. Used as a lead in to "escape sequences" see below.

1C ^\ Cursor left. The cursor is moved left one position. If it is already at the start of a line it is moved to the end of the previous line. If it is at the start of the display the code has no effect.

1D ^] Cursor right. The cursor is moved right one position. If it was at the end of a line it will move to the start of the next line. It will not move past the end-of-screen.

1E ^^ Cursor up. The cursor is moved up one line on the display. It will not move past the top line.

1F ^_ Cursor down. The cursor is moved down one line on the display. If it is on the bottom line of the display, the code will have no effect.

1B 25 [<ESC> %]
Clear to end-of-screen. The screen is cleared from the current cursor position.

1B 2A [<ESC> *]
Clear to end-of-line. The current line is cleared from the cursor position.

1B 3D RR CC [<ESC> = RR CC]
Cursor addressing. The cursor is positioned to row RR and column CC. The row and column addresses are offset by 20H and the top left hand corner of the screen has the

coordinates 0,0. ie to position to row 8, column 45 RR=20H+8 and CC=20H+45, giving a code sequence of - 1B 3D 28 4D.

4.3 On-Screen editing

Limited on-screen editing has been provided within the BIOS. It must be emphasised from the start that CP/M itself does not cater for this, and so this feature is a compromise, and if invoked at the wrong time can have some unexpected results.

The assumption made within the BIOS is that when this feature is invoked CP/M will be in its "buffered line input" mode. (ie the running program or the CCP has requested input via the BDOS function call 10, or handles input in a similar manner).

The edit mode is entered by typing a '1' while holding down the graphics key on the Nascom keyboard. (To change this character see section 5.2). The BIOS acknowledges that it has entered edit mode by changing the form of the cursor to a "←" on the Nascom screen, or a solid non-blinking cursor on the IVC card. The cursor control keys become operative, and the cursor can be positioned anywhere on the screen and the display corrected or changed. As in Nas-Sys shift/← and shift/→ will delete characters from a line, or enable new ones to be inserted. If it is desired to enter any control characters on the line being edited they must be entered with a leading ^ - ie exactly the way CP/M echoes a control character. (Note that this is only applicable within edit mode). If a control/Z is required in the line then ^ followed by Z should be typed. If the line being edited contains an up-arrow character then this should be preceed by another ^ to prevent the following character being converted into a control character. ie. ^^ is converted to a single ^ when enter is pressed.(In this case the line will shift down by a character position as the ^^ is condensed into a single ^).

Remember that if the line being edited is some way back up the screen, the screen can be partially cleared by moving the cursor to the end of that line and typing <ESCAPE> % (clear to end-of-screen, see section 4.2).

When "Enter" is typed the contents of the line currently holding the cursor is returned to CP/M and the edit mode terminates. The cursor is reset to its previous form.

The edit mode routine automatically leaves out any prompt at the start of a line from the character string it returns to CP/M, and so these do not have to be specifically edited out. This only applies to the following prompts:- A> B>...etc * - and #. Any others should be edited out.

4.4 Serial printer with handshake

A serial printer such as an IMP can be supported, either directly, or in a handshaking mode. The handshake signal is used to signify whether the printer is able to accept characters from the processor or not. It allows a printer to be connected by a high baud rate interface to ensure that it is driven at its maximum printing rate and never has to wait for

the processor. Without handshake the serial transmission rate has to be set to a low enough rate to ensure that the printer cannot be overdriven and characters lost.

The handshake signal must be a TTL level signal which should be connected to bit 7 of the Nascom keyboard port. (This is tp 3 on a Nascom 2). A low value on this line signifies that the printer is off-line or the buffer is full, and that it cannot accept characters. A high level signifies that characters can be sent to the printer.

The address of the UART supporting the printer is specified in the Patch Area (see section 5.2) and can be changed to suit.

4.5 Centronics type parallel printer

There is an in-built driver for a printer with a Centronics type parallel interface connected to the system via a PIO. The eight data wires of the interface should be connected to the B-side of the PIO, bit 0 to bit 0, and so on. The "busy" line should be connected to bit 0 of the A-side of the PIO, and the "strobe" line to bit 1 of the A-side. The "acknowledge" line is ignored.

Note that a standard Centronics interface includes pull-up resistors on the data and strobe lines in the printer. The PIO specification guarantees a low output voltage of 0.4V maximum when sinking 2ma (on both the A-side and the B-side). Although PIOs may apparently drive the printer interface directly, it might be advisable to include a TTL buffer between the PIO and printer if the pull-up resistors are of a low value.

The address of the PIO supporting the printer is specified in the Patch Area (see section 5.2) and can be changed to suit.

5. CUSTOMISATION

5.1 IOBYTE

The BIOS fully implements the IOBYTE function which allows logical to physical mapping of various drivers. The BIOS provides specific routines for I/O which CP/M calls. These support a Console, (Nascom Keyboard and screen in our case), Tape reader (cassette interface), Tape punch (cassette interface), and a List device (a printer of some kind).

What the IOBYTE does is to define which of various alternative routines should be used to service a particular request. The IOBYTE is held in memory at address 3 and is subdivided into four fields of two bits each as shown below:-

msb - L L P P R R C C - lsb

These four fields are LIST PUNCH READER CONSOLE

As each field consists of two bits it can take any one of four values. The purpose of the field is to define which of four alternative routines the function is to use. This is covered more fully below in the section on the Patch Area. When any of the four I/O routines are called the

first thing each does is to examine the relevant field of the IOBYTE and using the value found there execute one of four alternatives.

CP/M itself does not use the IOBYTE, but the STAT command can be used to temporarily change its value (see your CP/M manual).

5.2 Patch Area

To ease the problem of customisation (mainly for different printers) a "patch" area has been provided immediately following the jump table at the start of the BIOS. These values may be changed by using DDT or a similar program (eg ZSID) either to make a temporary change on a running system, or permanently on a system image just before "SYSGEN"ing a disk with it.

In the "SYSGEN" image the area can be found at 2133H irrespective of the system size. With a running system the jump address at 0 should be examined (use DDT again) to find the start of the BIOS area. For example if 0 holds JP 0B203H then the BIOS origin is at 0B200H and the patch area can be found at 0B233H. Hence XX in the following section should be replaced by B2.

Address	Name	Value	Purpose
XX33	IIOBYT	15	Initial value of the IOBYTE. The IOBYTE is reset to this value on Cold and Warm starts.
XX34	MS	xx	Loop counter used to obtain a delay of 1ms. Should be set to 4BH for a 4MHz system clock, 3CH for 4MHz + wait, and 26H for 2MHz.
XX35	BLINKR	10 04	(0410H) Count which sets the cursor blink rate. Decrease this number for a faster blink. If the IVC card is being used these two bytes define the cursor type to be used while the system is not waiting for keyboard input. (See the IVC software manual for details).It is intially set to a value that results in no cursor being displayed, ie. the cursor on the IVC screen behaves in a similar fashion to that on the Nascom screen.
XX37	INREP	00 50	(500H) Initial delay before the keyboard repeats a key. Decrease this number for a shorter delay.

XX39	RRPT	00 10	(100H) Keyboard running repeat rate. Decrease this number for a faster repeat.
XX3B	EMCHAR	B1	(Graphics+"1"). This character is the one that is recognised as the "enter screen-edit mode" command. Note that the keyboard can never return the character or control code selected for this. (The "GRAPH" key has been selected as the BIOS always returns characters to CP/M with bit 7 set to 0, and graphics characters as such cannot be returned. The "1" key is conveniently placed by the GRAPH key and is not affected by upper-case lock).

*****User I/O initialisation*****

The following three bytes can be set to a jump to a routine to initialise any special user i/o routines added to the system. This address is "Call"ed following a Cold start, but not after a Warm start.

XX3C	USRIOI	C9	RETurn
XX3D		00	NOP
XX3E		00	NOP

*****Port Addresses****

In the following section the base port addresses are defined for the serial routine. A common serial routine is used for all four fields (Console,Punch,Reader,List), and assumes a standard Nascom UART (ie 6402). However the lowest address of the UART is read from this area. This approach allows (for example) a high speed serial printer to be added to the system being driven through a UART on the Nascom IO board. All that is necessary from a software point of view to support this addition is to change the value of LISTP from 01 to the lowest address of the UART on the IO board, say 10H. The final address in this section of the table is that of a PIO if you wish the software to support a Centronics-type parallel printer. Once again the address to set is the lowest of the four that a PIO occupies.

XX3F	TTYP	01	TTY serial port address.
XX40	RDRP	01	Reader serial port address.
XX41	PUNP	01	Punch serial port address.
XX42	LISTP	01	List serial port address.
XX43	PPORT	04	List parallel port number. Set to 0 if no parallel printer.

****List Control****

In the following section a limited amount of enhanced support is provided for the list device if required. For example if you use a simple roll-feed printer that does not recognise form feeds you may find it advantageous to set FCNTRL to 6 to provide a small gap everytime a form feed occurs in the list stream.

XX44	FFCNTRL	00	Form feed control. If set to:- 00=Pass to printer unaltered. FF=Translate to multiple line feeds (see LPAGE and LGAP). nn=Translate to exactly nn linefeeds.
XX45	LPAGE	00	Lines per page. If set to 00=No paging. nn=nn printed lines per page.
XX46	LGAP	00	Gap between pages. If set to 00=no gap. nn=Gap of nn lines.

The following section holds seven tables, with each table consisting of four addresses. These four addresses are the addresses of the four alternative routines available for a particular IOBYTE field. Also shown is the name by which STAT refers to a particular field, and also the binary contents of the IOBYTE in the position of that field. For example if you wish to temporarily redefine the list device and use STAT LST:=TTY: to do it, then STAT will set bits 6 & 7 of the IOBYTE to 00. The list routine, finding its field (bits 6 & 7) set to 00, will use the routine whose address it finds at address XX77, ie SEROUT (serial out) in the unmodified case.

****IOBYTE control fields and address tables****

	Address	STAT ref.	IOBYTE field	Current routine used	
Console Status					
	XX47	TTY:	XXXXXX00	SERST	0=Serial
	XX49	CRT:	XXXXXX01	SCAN	1=Keyboard
	XX4B	BAT:	XXXXXX10	SERST	2=Serial
	XX4D	UC1:	XXXXXX11	SCAN	3=Keyboard
Console Input					
	XX4F	TTY:	XXXXXX00	SERIN	0=Serial
	XX51	CRT:	XXXXXX01	BLINK	1=Keyboard
	XX53	BAT:	XXXXXX10	SERIN	2=Serial
	XX55	UC1:	XXXXXX11	BLINK	3=Keyboard
Console Output					
	XX57	TTY:	XXXXXX00	SEROUT	0=Serial
	XX59	CRT:	XXXXXX01	CRT	1=Screen
	XX5B	BAT:	XXXXXX10	CRT	2=Screen
	XX5D	UC1:	XXXXXX11	CRT	3=Screen

```
Reader Input
          XX5F   TTY:  XXXX00XX   BLINK   0=Keyboard
          XX61   PTR:  XXXX01XX   SERIN   1=Serial (Cassette)
          XX63   UR1:  XXXX10XX   BLINK   2=Keyboard
          XX65   UR2:  XXXX11XX   BLINK   3=Keyboard

Punch Output
          XX67   TTY:  XX00XXXX   CRT     0=Screen
          XX69   PTP:  XX01XXXX   SEROUT  1=Serial (Cassette)
          XX6B   UP1:  XX10XXXX   CRT     2=Screen
          XX6D   UP2:  XX11XXXX   CRT     3=Screen

List Status
          XX6F   TTY:  00XXXXXX   SERST   0=Serial
          XX71   CRT:  01XXXXXX   READY   1=Screen
          XX73   LPT:  10XXXXXX   HSST    2=Serial with Handshake
          XX75   UL1:  11XXXXXX   CENTST  3=Centronics parallel

List Output
          XX77   TTY:  00XXXXXX   SEROUT  0=Serial
          XX79   CRT:  01XXXXXX   CRT     1=Screen
          XX7B   LPT:  10XXXXXX   HSHAKE  2=Serial with Handshake
          XX7D   UL1:  11XXXXXX   CENTRO  3=Centronics parallel
```

**********User Area**********

The following a 32-byte area is free and provides an area where a small user routine may be patched in. The appropriate address above should be changed to point to this area.

```
          XX7F    USER:   DEFS    32
```

APPENDIX 1 SIMON

SIMON is a simple monitor that resides in the Boot EPROM. His sole purpose is to provide a debugging aid in the event of problems with the hardware. Obviously if the problem is a severe one SIMON himself will not run. It is to be hoped that his services will never be required. He supports both the normal Nascom screen and the Gemini IVC card, but only the normal Nascom keyboard.

SIMON expects input from the keyboard, automatically converting all alphabetic characters to upper case. He is not forgiving of typing errors, (backspace is not recognised), and the only acceptable control code is carriage return.

> When SIMON is waiting for input he puts out the prompt '>>'. All inputs to SIMON consist of a command letter followed by one or more hexadecimal arguments separated from each other by spaces, and terminated by a carriage return.

The commands are summarised below and then explained in greater detail in the following section.

B Boot disk
C Copy memory
E Execute from an address
F Fill memory with a byte
M Modify memory
O Output to a port
Q Query a port
T Tabulate memory contents

COMMANDS

B

Boot from disk. This command does not wait for a <carriage-return> before executing. It results in a jump being made to the start of the EPROM and is equivalent to pressing the RESET key, and so another attempt will be made to Boot CP/M in from disk.

C <start> <destination> <length>

This command copies the number of bytes specified in <length> from address <start> to address <destination>. The bytes are copied by the LDIR instruction, and so due care must be exercised if it is desired to copy intact between two areas of memory that overlap.

E <address>

This command results in a CALL to <address>. This enables manually entered test routines to be executed. Return to SIMON can simply be effected by a RETurn.

F <start> <end> <value>

This command fills the memory from <start> to <end> inclusive with the byte <value>.

M <address>

This command allows the contents of memory to be displayed and optionally changed. SIMON will respond with:-
>>M1000<CR> (Command typed in response to prompt)
1000 - FE _← Cursor

Typing a carriage-return will leave the contents unaltered and the next location will be displayed. If it is desired to change the contents then the new value should be typed followed by a space or carriage-return. Any non-hex character typed will cause the command to abort without the current location being modified. When a location is modified SIMON checks to ensure that the new value has been written successfully. If it has not then the command aborts with the error message -What? . This also occurs if an attempt is made to enter a number greater than OFFH.

Typing a '-' causes SIMON to move back to the previous address. This allows any erroneous value to be modified without having to terminate the command and restart it.

O <port> <value>

Outputs the number <value> to port address <port>.

Q <port>

Inputs from port address <port> and displays in Hex the byte fetched.

T <address> <lines>

Displays the contents of memory in Hex starting at address <address>. 8 consecutive bytes are displayed per line, with the address of the first byte being the first item on the line. The command terminates when <lines> lines have been displayed.

Several vectors into the EPROM are provided for those writing short test routines. These addresses should be CALLed to provide the required function.

F000	COLD	Cold start
F003	CHRIN	Get a character from the keyboard. All lower-case characters are converted to upper-case, and the character is echoed to the screen.
F006	CHROUT	Output a character to the screen.
F009	P2HEX	Print <A> as two Hex characters
F00C	P4HEX	Print <HL> as four Hex characters
F00F	SPACE	Print a space
F012	CRLF	Print a carriage-return

APPENDIX 2 CUSTOMISATION EXAMPLE

For the purpose of this example let us assume that it is desired to add support for a printer such as an IBM golfball or a Creed teleprinter, both of which require special drivers to handle them.

The necessary drivers can be written,assembled and tested within CP/M using the software tools available. As these drivers will be fairly large there is no room for them within the standard BIOS, but an obvious place for them is in the 1k of EPROM space above SIMON. ie in the area 0F400H-0F7FFH. As the BIOS uses no workspace outside the running CP/M system the workspace RAM from 0FC00H to 0FFFFH can be used by the driver. To ease the problem of later updates it would be advisable to start the EPROM with three Jumps to the routines required:-

```
F400  JP  INIT      ;Printer initialisation
F403  JP  STATUS    ;Printer status
F406  JP  OUTPUT    ;Printer character output
```

INIT may or may not be required. In the cold start section of the BIOS a CALL is made to USRIOI (see section 5.2) which is initially set to a RET. This can be redirected to a routine to initialise the new printer driver, setting up its workspace appropriately.

STATUS returns the status of the printer in the A register (see the CP/M alteration guide manual). Register A is set to 00 if the printer is busy and unable to receive another character, and set to 0FFH if it is ready and can receive another character.

OUTPUT outputs the character in register C to the printer.

If any of the routines use more than eight levels of stack, it is advisable to set up a local stack pointer for use within the routine, and to reset it before returning.

Once the printer driver has been written, tested and placed in EPROM the following steps should be taken to integrate it into the CP/M system.

For this example we will assume a 48k system is required for a standard Nascom 2 running at 4MHz without wait states, and without the Gemini IVC. (For the IVC use MOVCPMV!).

All user input is underlined, and a carriage-return is represented by <CR>. (Where you see <CR> you should type ENTER).

First a 48k system has to be generated and saved on the disk.

```
A>movcpmn 48 *<CR>

Constructing 48k CP/M vers 2.2
Ready for "SYSGEN" or
"SAVE 44 CPM48.COM"
A>save 44 cpm48.com<CR>
```

```
A>
```

Next use DDT (or a similar program) to load the image back into memory and to patch it.

```
A>ddt cpm48.com<CR>
DDT VERS 2.2
NEXT  PC
2D00 0100
```

The patch area starts at 2133H (see section 5.2). First change USRIOI to point to the EPROM.

```
-s213c<CR>
213C C9 c3<CR>     (Change RET to JP)
213D 00 <CR>       (Address wanted is 0F400H)
213E 00 f4<CR>
213F 01 .<CR>      (Stop the S command).
-
```

Let us select the LPT: option of the list field to point to the new driver. First change IIOBYT to reflect this.

```
-s2133<CR>
2133 15 95<CR>     (LPT field is 10......)
2134 4B .<CR>      (Stop the S command)
-
```

Finally modify the table entries in the list section to point to the new routines. (Note the current contents of the addresses below may be different in your case).

```
-s2173<CR>
2173 D5 03<CR>     (Address 0F403H)
2174 B7 f4<CR>
2175 01 .<CR>
-s217b<CR>
217B EF 06<CR>     (Address 0F406H)
217C B7 f4<CR>
217D 0B .<CR>
-
```

The patching is now complete, so exit from DDT with a control/C (or G0).

```
A>
```

Now save the modified image back to disk.

```
A>save 44 cpm48.com<CR>
A>
```

Finally put the new system on the system tracks of a disk using SYSGEN. (Here we've selected drive B).

```
A>sysgen cpm48.com<CR>
SYSGEN VER 2.0
Destination Drive name (or Return to reboot)b  (or a)
Destination on B, then type Return
```

At this stage a disk can be placed in the selected drive and "Return" typed.

```
**Function complete**
Destination Drive name (or Return to reboot)
```

Further disks can be "sysgen"ed, or the new disk placed in drive A and Reset pressed in order to try it out.

APPENDIX 3 UTILITIES

FORMAT

This program will format or verify disks in single or double density. It contains its own in-built disk drivers and does not use any of the BIOS routines for disk access.

On running it, it will clear the screen and ask for the density required. Reply with S for single density, and D for double density. If double density has been selected it will then prompt for the skew required (see section 2.5 for an explanation of skew). A skew of 2 is suggested for a 4MHz system, but experiment to see which suits your system and application. Too small a skew will increase the delay between typing ^C and getting the CP/M prompt, though it may well speed up the loading of .COM type files. Too large a skew will slow the disk system down unnecessarily.

Following this a prompt for the drive to use will appear. Type the desired drive number.

If desired the format program can be used to just verify that a disk can be read in its entirety without error. This is done by typing "V" at this point. Typing "F" will format the disc, after which it will automatically verify the disk. Following the formatting sequence, and before verifying the disk, the program comments on the average drive rotational speed. It will either say "Average drive speed Ok", or "Average drive speed out of limits". In the latter case the drive speed has been found to be more than 1.5% in error, and steps should be taken to have this corrected.

During the verification phase the software will report any time it has to re-read a sector, and also displays the error type returned by the disk controller. After four retries the head is restored to track 0 and the software tries again. Once the retry count reaches eight that sector is considered unreadable and is abandonded, the software moving on to the next sector. Ideally no retries should occur, but possibly one or two might. Any disk that produces a large amount of retries should be regarded with suspicion, and any one that has an unreadable sector should not be used.

BACKUP

This utility has been provided as a means of making backup copies of disks. PIP can obviously be used for this purpose, (eg PIP B:=A:*.*[V] to backup all files on drive A to drive B), and has the advantage that it repacks the files on the disk on the way, but it can take a long time to copy a large number of files. Also a separate program (SYSGEN) has to be used to copy the CP/M system track across.

BACKUP however includes its own disk drivers and physically copies a direct image of the source disk to the destination disk (including the system track) and then verifies it. The copy procedure terminates when the end of the disk is reached, or when it reads a entire buffer of 0E5Hs indicating that CP/M has not written any data past that point. (The Format program fills all sectors with 0E5H). There is no

blocking/deblocking and no CP/M overhead of opening and closing files. Hence it only takes a few minutes to make an up-to-date copy of a working disk. It also provides an easy way for people with single drive systems to backup their disks.

On running BACKUP it prompts for the source and destination drives. It is permissible to enter the same drive letter for both drives. In this case BACKUP will pause before loading the buffer from the source disk and will ask for it to be inserted. Similarly it will ask for the destination disk to be inserted before writing the buffer out. This will be repeated until the source disk is copied. The larger the size of the running CP/M system (and hence BACKUPs buffer), the sooner the copy will be completed. In the single drive case it is advisable to Write Protect the source disk with a tab to guard against inserting it at the wrong time.

The program only works in double density.

APPENDIX 4 HARDWARE CHECK

This appendix lists a few commands that can be given within SIMON to confirm that the disk interface is working. It is not intended to be an exhaustive check, but will indicate if there are any major problems.

Following a RESET the command QE0 should return 00. Insert a disk (preferably formatted for double density) and follow the following sequence of commands.

Command	Reason	Effect
OE4 01	Select Drive and start motors. Use 01 for drive A, 02 for drive B, 04 for drive C and 08 for drive D.	Drive motors should start if not already going.
OE0 0B	Restore head on selected drive.	Head should load on selected drive and restore to track 00
QE0	Check status. Should be done while head is still loaded. This should return 24. (Head loaded & track 00). It may also have bit 6 set (if a write-protect tab is fitted), and bit 1 (Index pulse).	The head will unload automatically six revolutions after the command completes.
OE3 20	Write 20H (32) into the data register.	
OE4 01	Keep motors going.	Restarts the motors if they have stopped
OE0 1F	Issue SEEK command with verify. (If disk is not formatted then use 1B).	Head should load and move to track 32.
QE0	Check status, should be done while the head is still loaded. Should return 20. (possibly 60 if write-protect tab fitted). Bit 4 set if seek error, (correct header record not found). Bit 0 set if command still in progress.	